ICS 35.240.01
CCS L 70

# T/SZSSIA

团　　体　　标　　准

T/SZSSIAXXX—2021

# 算法仓管理系统通用技术要求

General technical requirements of algorithm warehouse management system

（征求意见稿）

2021-06-24

**在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上。**

XXXX-XX-XX 发布　　XXXX-XX-XX 实施

深圳市智慧安防行业协会　发布

# 目　　次

# 前　言

本文件按照GB/T 1.1—2020《标准化工作导则第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由深圳云天励飞技术股份有限公司提出。

本文件由深圳市智慧安防行业协会归口。

本文件起草单位：

本文件主要起草人：

# 算法仓管理系统通用技术要求

## 1 范围

本文件规定了算法仓管理系统的设计原则、系统架构、业务流程和功能、接口、性能等技术要求。

本文件适用于算法包集中存储、管理、调度及事件检测的管理系统的设计、开发。

## 2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 28181 公共安全视频监控联网系统信息传输、交换、控制技术要求

GA/T 1400.4—2017 公安视频图像信息应用系统第4部分：接口协议要求

## 3 术语和定义

下列术语和定义适用于本文件。

3.1

**算法包 algorithm package**

可对输入数据进行推理运算的算法模型程序集合。

注：输入数据为结构化信息，或视频、图像、图片、语音等非结构化信息。

3.2

**算法仓 algorithm warehouse**

用于集中存储、管理若干算法包的仓库。

3.3

**算法仓管理系统 algorithm warehouse management system**

对算法仓进行统一鉴权、算法包版本管理、算法包增删等，对数据源采用算法仓中的算法进行事件检测，并输出检测结果的系统。

注：在不引起混淆的情况下，本文件中的“算法仓管理系统”简称为“系统”。

3.4

**订阅者 subscriber**

算法仓分析事件的上报对象。

## 4 缩略语

下列缩略语适用于本文件。

API：应用程序接口（Application Programming Interface）

CPU：中央处理单元（Central Processing Unit）

GPU：图形处理单元（Graphics Processing Unit）

JSON：JS对象简谱（JavaScript Object Notation）

ONVIF：开放网络视频接口论坛（Open Network Video Interface Forum）

NPU：神经网络处理单元（Neural-Network Processing Unit）

REST：表述性状态转移（REpresentational State Transfer）

RTSP：实时流化协议（Real-TimeStreamingProtocol）

## 5 系统设计原则

系统的设计应遵循以下原则：

a) 开放性：应满足应用平台和算法之间解耦，保证数据、算力、算法、应用等各层能力的开放；
b) 可靠性：系统连续运行，各项功能正常；
c) 可扩展性：架构可扩展，性能支持平滑扩容，可适应相应场景管理需求的增加。

## 6 系统架构和业务流程

### 6.1 系统架构

#### 6.1.1 系统组成

系统由数据源接入模块、算法仓模块、算法仓管理模块等组成，系统架构示意图见图1。

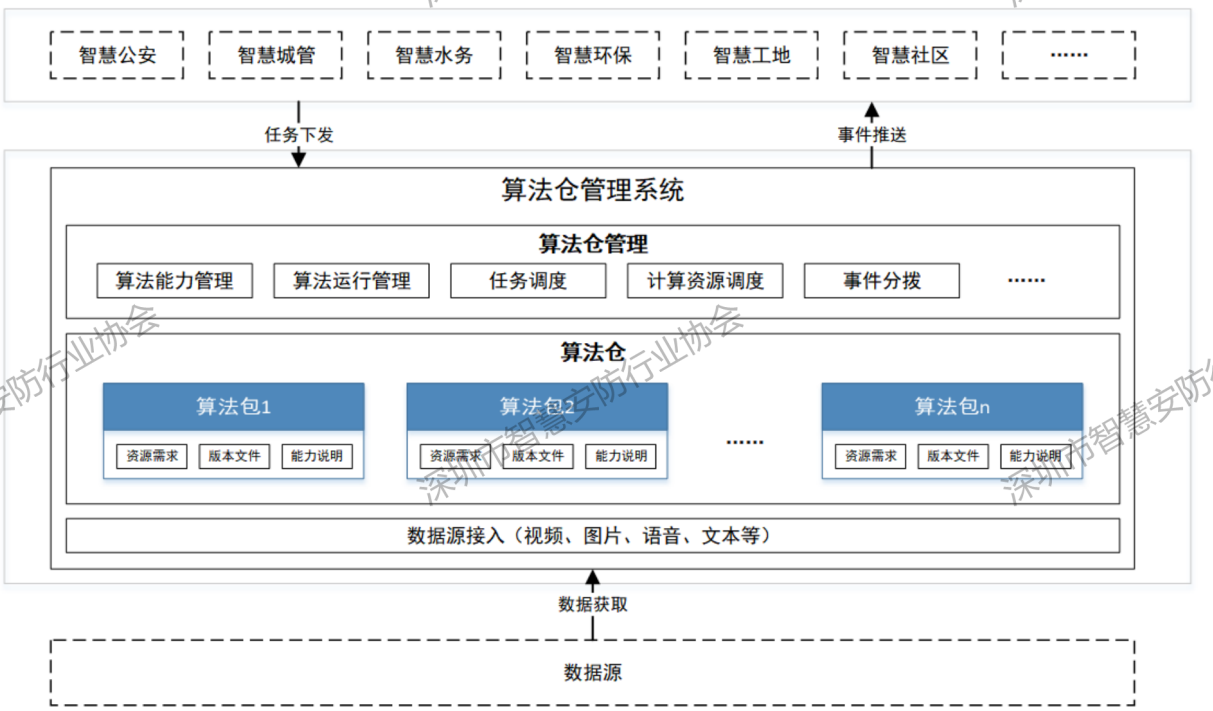


图1 系统架构示意图

#### 6.1.2 数据源接入

用于接入外部数据源，包括但不限于视频、图片、语音、文本等数据。

#### 6.1.3 算法仓

算法仓中包含若干算法包，算法包对接入的数据源进行事件分析，并上报分析结果。每个算法包包含若干必要的组件，应包括但不限于：

a) 资源需求；
b) 版本文件；
c) 能力说明。

#### 6.1.4 算法仓管理

管理算法包的运行状态，对算法算力进行必要的管理，并与算法仓行数据交互。系统应具备对算法仓运行状态的管理功能，该功能应包括但不限于：

a) 算法能力管理：用于获取算法仓中所有算法包的能力接口，形成算法能力列表；
b) 算法运行管理：用于对算法包的运行状态进行管理，包括注册状态、心跳状态等；

c) 任务调度：用于给算法仓下达任务，并支持任务状态的查询；
d) 计算资源调度：对计算资源进行调度，以满足算法包的计算资源要求；
e) 事件分拨：用于对算法包分析结果进行分类，并路由到相应的订阅者。

## 6.2 业务流程

系统运行的业务流程主要包括算法部署启动、任务调度、资源调度、事件分拨等。系统的常规业务流程见图2，具体如下：

a) 任务调度模块接收到分析任务，包含数据源、执行时间、算法种类等信息；
b) 算法能力管理模块依据算法种类获取得到算法包，获取得到相应的资源需求信息；
c) 计算资源调度模块依据资源需求信息，分配获取得到对应的计算资源；
d) 算法运行模块依据任务信息，将任务指定的算法包运行在分配的计算资源上；
e) 算法包依据任务信息，通过数据接入模块获取分析数据源，经过计算分析得到事件结果，反馈给事件分拨模块；
f) 事件分拨模块按照事件类型分拨给对应的订阅者，完成分析任务闭环。

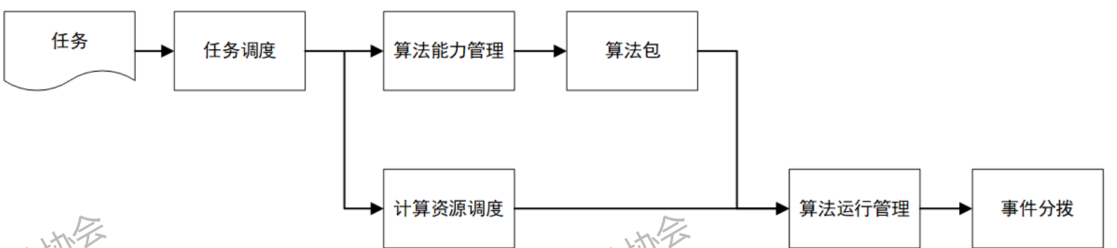


图2 业务流程图

# 7 系统技术要求

## 7.1 功能要求

### 7.1.1 数据源接入

数据源接入应具备以下功能：

a） 通过 GB/T 28181、RTSP、ONVIF 或其他私有协议对接视频采集设备、视频采集系统；
b） 通过 GA/T 1400.4 或其他私有协议对接图片采集系统；
c） 通过接口方式传入视频文件，支持 mp4、mpg 等文件格式；
d） 通过接口方式传入语音，支持 pcm、wav、mp3、mp4 等格式的音频文件；
e） 通过接口方式传入文本等结构化数据。

### 7.1.2 算法仓

算法仓支持系统的管理，应具备以下功能：

a) 能力查询模块：支持查询获取算法包的能力接口；
b) 运行状态：用于像上级管理平台执行注册、心跳等运行健康状态维护逻辑；
c) 任务状态：对于所执行的分析任务反馈执行进度等状态信息；
d) 事件分析：对数据源，依据下发的任务执行指定的检测程序，并返回分析结果；
e) 日志记录：记录算法包运行、操作、执行的记录，在问题诊断追踪、理解系统运行活动时使用。

### 7.1.3 算法能力管理

算法能力管理模块支持对算法能力信息的获取，应包括但不限于：

a） 算法供应厂家；
b） 算法的功能描述；
c） 算法版本信息；

d） 算法功能标签及场景匹配标签；
e） 算法的 CPU、内存、NPU/GPU 等计算资源需求及性能数据；
f） 算法的 API 接口列表及参数描述。

#### 7.1.4 算法运行管理

算法运行管理主要完成对算法的运行状态管理，应具备以下功能：
a） 支持算法注册、注销服务；
b） 支持算法心跳应答服务；
c） 支持算法重启服务；
d） 支持算法运行状态管理服务。

#### 7.1.5 任务调度

任务调度具备对算法仓下发任务以及任务的管理功能，应具备以下功能：
a） 任务应包含数据源、执行时间、算法种类信息；
b） 支持查询任务状态；
c） 支持任务暂停、重启、停止等。

#### 7.1.6 资源调度

资源调度具备计算资源的查询分配功能，应具备以下功能：
a） 查询计算资源的状态；
b） 按需指定任务执行的具体计算资源。

#### 7.1.7 事件分拨

事件分拨支持外部系统对事件的订阅以及事件分发，应具备以下功能：
a） 支持业务端按类型订阅分析事件；
b） 支持对算法分析事件进行记录、分类统计；
c） 支持根据事件类型分别发送给不同的订阅者。

### 7.2 接口要求

#### 7.2.1 总体要求

系统内部接口协议形式应采用RESTful，数据格式应按照JSON格式定义。

#### 7.2.2 默认信息端口

算法仓应以25030端口作为默认信息端口，并在这个端口上进行监听，提供默认基本接口调用服务。

#### 7.2.3 健康状态检查

此接口用于测试算法仓的健康状态，此接口检测正常应答，则认为算法仓处于正常运行状态，接口对传入的字符串进行复制及返回，见表1。

表1 健康状态检查

| 功能 | 对算法仓进行健康状态检测 | | | |
|---|---|---|---|---|
| 承载协议 | HTTP(POST) | | | |
| 请求的URI | /api/common/echoback | | | |
| 请求的Header | Content-Type:application/json;charset=UTF-8 | | | |
| 请求消息字段说明 | | | | |
| 名称 | 说明 | 数据类型 | 是否必选 | 备注说明 |
| CanYouHear | 服务接口在返回时复制的字符串 | String | 是 | 无 |
| 请求消息示例 | | | | |

```
{
    “CanYouHear”:“Message just for test!!”
}
```

返回消息字段说明

| 名称 | 说明 | 数据类型 | 是否必选 | 备注说明 |
|---|---|---|---|---|
| RogerThat | 服务接口调用时按“CanYouHear”原样进行返回 | String | 是 | 无 |

返回消息示例

```
{
    “RogerThat”:“Message just for test!!”
}
```

### 7.2.4 算法信息查询

算法仓可通过默认信息端口提供查询服务，支持对算法仓算法详细信息进行查询，见表2。

**表2 算法仓信息查询**

| 功能 | 查询算法仓的基本信息 | | | |
|---|---|---|---|---|
| 承载协议 | HTTP(POST) | | | |
| 请求的URI | /api/common/baseinfo | | | |
| 请求的Header | Content-Type:application/json;charset=UTF-8 | | | |
| 请求消息字段说明 | | | | |
| 名称 | 说明 | 数据类型 | 是否必选 | 备注说明 |
| 无 | 无 | 无 | 无 | 无 |
| 返回消息字段说明 | | | | |
| 名称 | 说明 | 数据类型 | 是否必选 | 备注说明 |
| Vender | 厂商名称 | String | 是 | 厂商或公司名称或简称，同一个公司发布的算法包这个字段需要保持一致 |
| Version | 版本号 | String | 是 | 版本号信息，指算法包的版本号 |
| CpuArch | 适用的CPU架构 | String | 是 | 可选的选项有“X86”，“AMD64”，“ARMv8”等选项，其他选项参考样式即可 |
| CpuNick | 适用的CPU别称 | String | 否 | 所主要适配的处理器的厂商的系统器系列，如“XEON E5”，“KunPeng920”，“FeiTeng 2000+”等处理器，其他选项参考样式即可 |
| AccType | 推理计算加速类型 | String | 是 | 典型取值有“GPU”，“NPU”，非推理计算加速类型取值“CPU” |
| AccVersion | 推理加速芯片或加速卡的版本 | String | 是 | 典型取值有“Nvidia P4”，“Nvidia T4”，“Atlas 300”，“DeepEye 1000”等，其他选项参考样式即可 |
| DriverVersion | 推理加速卡等芯片的版本 | String | 是 | 典型取值有“Nvidia 418.37”，“Atlas C30B902”等版本号，其他选项参考样式即可 |
| LibVersion | 基于推理加速卡的开发库版本 | String | 是 | 典型取值有“Nvidia CUDA 10.0”，“Atlas 1.0/2.0”，“DESDK 1.0”等版本号，其他选项参考样式即可 |
| ResourceReq | 资源需求 | Object | 是 | 资源需求说明 |
| ResourceReq.Cpu | 需要的CPU资源，单位为 | Int | 是 | 在算法仓启动时候，由系统进行分配 |

| | | | | |
|---|---|---|---|---|
| | CPU核心数 | | | |
| ResourceReq.Mem | 需要的内存的大小，单位为M | Int | 是 | 在算法仓启动时候，由系统进行分配 |
| ResourceReq.Acc | 需要的推理加速卡的数量 | Int | 是 | 在算法仓启动时候，由系统进行分配 |
| 返回消息示例 | | | | |


```
{
“Vender”:“Intellifusion”,
“Version”:“v3.3.5”,
“CpuArch”:“ARMv8”,
“CpuNick”:“KunPeng920”,
“AccType”:“NPU”,
“AccVerion”:“Atlas 300”，
“DriverVersion”:“C30B902”,
“LibVersion”:“DDK 1.0”，
“ResourceReq”:{
        “Cpu”:32,
        “Mem”:30000,
        “Acc”:2
        }
}
```


### 7.2.5 算法能力查询

算法仓可通过默认信息端口提供查询服务，支持对算法仓算法能力信息进行查询，见表3。

**表3 算法能力查询**

| 功能 | 读取算法仓的算法能力描述信息等 | | | |
|---|---|---|---|---|
| 承载协议 | HTTP(POST) | | | |
| 请求的URI | /api/common/capacity | | | |
| 请求的Header | Content-Type:application/json;charset=UTF-8 | | | |
| 请求消息字段说明 | | | | |
| 名称 | 说明 | 数据类型 | 是否必选 | 备注说明 |
| 无 | 无 | 无 | 无 | 无 |
| 返回消息字段说明 | | | | |
| 名称 | 说明 | 数据类型 | 是否必选 | 备注说明 |
| CapacityNum | 接口或功能数量 | Int | 是 | 算法包所包含的接口或功能数量，与后续的描述一致 |
| CapacityInfo | 接口或功能的信息 | Object[] | 是 | 版本号信息，指算法包的版本号 |
| CapacityInfo[].Vender | 厂商名称 | String | 是 | 厂商或公司名称或简称，同一个公司发布的算法包这个字段需要保持一致，用于归类算法 |
| CapacityInfo[].ShortName | 接口所涉及的算法简称 | String | 是 | 如“CrowdDetect”，“FlagDetect”等清晰简洁的缩写，用于归类算法。 |
| CapacityInfo[].Port | 接口的端口 | String | 是 | 无 |
| CapacityInfo[].Uri | 接口的URL相对路径 | String | 是 | 无 |
| CapacityInfo[].InfType | 接口调用类型 | String | 是 | 主要取值的单次同步类型(OnceSync)，单次异步类型(OnceAsync)，无连接任务异步类型(SyncTask)，有连接任务异步类型(SyncConnTask) |

| | | | | |
|---|---|---|---|---|
| CapacityInfo[].AsyncType | 如果为异步，异步返回数据形式 | String | 是 | 主要取值的Http回调(HttpCallback)，Kafka内部消息队列回调(KafkaCallBackInner)，Kafka外部消息队列回调(KafkaCallBackOuter)，异步方式填null |
| CapacityInfo[].Tps | 接口调用的吞吐量性能，即单位时间内完成的调用次数 | String | 是 | 典型如单位时间处理图片数量，或可支持并发视频分析路数 |
| CapacityInfo[].Delay | 接口调用的时延性能，即完成一次调用的所需时间 | String | 否 | 对于基于连接的异步任务类型，由于没有完成的概念所以也就没有时延数据 |
| CapacityInfo[].InputData | 接口调用输入数据的类型 | String | 是 | 典型的取值有“Image”，“Video”，“Text” |
| CapacityInfo[].Description | 接口功能描述说明 | String | 是 | 无 |
| 返回消息示例 | | | | |

```
{
“CapacityNum”:1,
“CapacityInfo”:[
        {
        “Vender”:“Intellifusion”,
                “ShortName”:“AerialParabolic”,
        “Port”:8080,
        “Uri”:“Algapi/AerialParabolic”,
        “InfType”:“SyncConnTask”,
        “AsyncType”:“HttpCallback”
        “Tps”:16,
        “InputData”:“Video”,
        “Description”:“A CNN Alg Module to Detect weather Parabolic aerial phenomenon happend”
        }
    ]
}
```

### 7.2.6 服务端口监听

算法仓采用多端口监听策略，其具体的端口列表可通过算法能力查询接口进行获取。

### 7.2.7 服务心跳

对于算法仓，需要定期向系统主动报告自己的状态，称之为算法仓的心跳上报，以此驱动资源均衡、动态服务路由及高可用策略等。上报采用回调的机制，算法仓启动时先通过以下环境变量获取回调的IP/端口/URL：

a) HEATBEAT_CALLBACK_IP，心跳上报的远程服务器的 IP 地址；
b) HEATBEAT_CALLBACK_PORT，心跳上报的远程服务器的端口；
c) HEATBEAT_CALLBACK_URL，心跳上报的远程服务器接口的相对 URL；
d) HEATBEAT_CALLBACK_TYPE，心跳上报的自身算法类型编码；
e) HEATBEAT_CALLBACK_RATE，心跳上报的频率，间隔秒数。

系统的任务调度模块提供了如下标准的心跳上报回调接口，见表4。

**表4 服务心跳**

| 功能 | 服务心跳接口 |
|---|---|
| 承载协议 | HTTP(POST) |
| 请求的URI | 通过HEATBEAT_CALLBACK_URL环境变量获取 |
| 请求的Header | Content-Type:application/json;charset=UTF-8 |
| 请求消息字段说明 | |

| 名称 | 说明 | 数据类型 | 是否必选 | 备注说明 |
|---|---|---|---|---|
| uuid | 节点uuid值，用以表示当前算法仓实例的唯一身份 | String | 是 | 启动时自生成，不重启uuid不变 |
| ip | 节点ip值，用以返回算法仓实例ip | String | 是 | 来自于算法包配置文件定义规范的env环境变量配置 |
| port | 节点port值，用以返回算法仓的实例的算法的端口 | String | 是 | 无 |
| type | 算法仓的的类型编码 | Int | 是 | 由中心平台统一编码分配 |
| version | 算法包版本，由厂商自行维护提供 | String | 是 | 无 |
| tasks | 算法算法仓正在运行的任务 | Object[] | 否 | 无 |
| Tasks[].uuid | 算法任务uuid的值，注意是由外部任务的请求方传入 | string | 是 | 无 |
| Tasks[].algPlatformData | 由用户传入需要在回调时回传的数据 | String | 是 | 无 |
| Tasks[].taskParam | 算法任务参数 | String | 是 | 无 |
| Tasks[].status | 算法任务执行状态 | Integer | 否 | 默认为0（0代表正常，1代表异常） |
| 返回消息字段说明 | | | | |
| 名称 | 说明 | 数据类型 | 是否必选 | 备注说明 |
| 无 | 无 | 无 | 无 | 无 |
| 请求消息示例： | | | | |


```
{
    "uuid": "4cdbc040-657a-4847-b266-7e31d9e2c3d9",
    "ip": "192.168.100.119",
    "port": "8080",
    "type": 10001,
    "version": "3.2.11",
    "tasks": [
        {
            "uuid": "72297c88-4260-4c05-9b05-d28bfb11d10b",
            "algPlatformData": "admin@算法标识",
            "taskParam":
"{\"videoType\":\"liveVideo\",\"videoUrl\":\"rtsp://192.168.11.210/live\",\"uuid\":\"ddb366f5-d4bc-3a20-ac68-e13c0560058f\",\"snaperRoi\":[{\"mode\":1,\"pointList\":[{\"x\":50,\"y\":0},{\"x\":100,\"y\":0},{\"x\":100,\"y\":80},{\"x\":50,\"y\":80}]}],\"userData\":\"intellif\",\"algPlatformData\":\"lg@2204\",\"condition\":\"{\\\"confidence\\\":0.9}\",\"dataNotifyUrl\":\"http://127.0.0.1:8080/api/task/dataNotify\",\"debugEnable\":true}",
            "status": 0
        }
    ]
}
```


### 7.2.8 服务注册 CallBack

算法仓启动后，应能够在系统上注册算法，并发送算法的各类能力信息。注册采用回调的机制，算法仓启动时先通过以下环境变量获取注册回调的IP、端口、URL等信息。

a) REGISTER_CALLBACK_IP，算法注册信息上报的远程服务器的 IP 地址；
b) REGISTER_CALLBACK_PORT，算法注册信息上报的远程服务器的端口；
c) REGISTER_CALLBACK_URL，算法注册信息上报的远程服务器接口的相对 URL；
d) REGISTER_CALLBACK_TYPE，算法注册信息上报的远程服务算法仓类型编码。

系统的算法运行管理模块，提供了如下标准的注册回调接口，见表5。

**表5 服务注册**

| 功能 | 服务能力注册的CallBack | | | |
|---|---|---|---|---|
| 承载协议 | HTTP(POST) | | | |
| 请求的URI | 通过REGISTER_CALLBACK_URL环境变量获取 | | | |
| 请求的Header | Content-Type:application/json;charset=UTF-8 | | | |
| 请求消息字段说明 | | | | |
| 名称 | 说明 | 数据类型 | 是否必选 | 备注说明 |
| uuid | 节点uuid值，用以表示当前算法仓实例的唯一身份 | String | 是 | 启动时自生成，不重启uuid不变 |
| ip | 节点ip值，用以返回算法仓的实例的ip | String | 是 | 来自于算法包配置文件定义规范的env环境变量配置 |
| type | 节点算法仓类型编号 | Int | 是 | 由中心平台统一编码分配 |
| CapacityNum | 接口或功能数量 | Int | 是 | 算法包所包含的接口或功能的数量，与后续的描述一致 |
| CapacityInfo | 接口或功能的信息 | Object[] | 是 | 算法包所包含的接口或功能的数量，与后续的描述一致 |
| CapacityInfo[].Vender | 厂商名称 | String | 是 | 厂商或公司名称或简称，同一个公司发布的算法包这个字段需要保持一致，用于归类算法 |
| CapacityInfo[].ShortName | 接口所涉及的算法简称 | String | 是 | 如“CrowdDetect”，“FlagDetect”等清晰简洁的缩写，用于归类算法。 |
| CapacityInfo[].Port | 接口端口 | String | 是 | 无 |
| CapacityInfo[].Uri | 接口URL相对路径 | String | 是 | 无 |
| CapacityInfo[].InfType | 接口调用类型 | String | 是 | 主要取值的单次同步类型(OnceSync)，单次异步类型(OnceAsync)，无连接任务异步类型(SyncTask)，有连接任务异步类型(SyncConnTask) |
| CapacityInfo[].AsyncType | 如果为异步的话，异步返回数据形式 | String | 是 | 主要取值的Http回调(HttpCallback)，Kafka内部消息队列回调(KafkaCallBackInner)，Kafka外部消息队列回调(KafkaCallBackOuter) |
| CapacityInfo[].Tps | 接口调用的吞吐量性能，即单位时间内完成调用的次数 | String | 是 | 典型如单位时间处理的图片的数量，或可支持的并分视频分析路数 |
| CapacityInfo[].Delay | 接口调用的时延性能，即完成一次调用的所需要的时间 | String | 否 | 对于基于连接的异步任务类型，由于没有完成的概念所以也就没有时延数据 |
| CapacityInfo[].InputData | 接口调用输入数据的类型 | String | 是 | 典型的取值有“Image”，“Video” |
| CapacityInfo[].Description | 接口功能的描述与说明 | String | 是 | 无 |
| 返回消息字段说明 | | | | |
| 名称 | 说明 | 数据类型 | 是否必选 | 备注说明 |
| 无 | 无 | 无 | 无 | 无 |
| 请求消息示例： | | | | |

```
{
      "uuid": "4cdbc040-657a-4847-b266-7e31d9e2c3d9",
      "ip": "192.168.100.119",
“CapacityNum”:1,
“CapacityInfo”:[
          {
             “Vender”:“Intellifusion”,
             “ShortName”:“AerialParabolic”，
“Port”:8080,
“Uri”:“Algapi/AerialParabolic”,
“InfType”:“SyncConnTask”,
“AsyncType”:“HttpCallback”
“Tps”:16，
“InputData”:“Video”,
“Description”:“A CNN Alg Module to Detect weather Parabolic aerial phenomenon happend”
          }
      ]
}
```


## 7.3 性能要求

### 7.3.1 心跳周期

算法仓和算法之间以心跳来维持相互存在状态的感知，符合以下要求：

a） 心跳周期不超过 5 秒；

b） 超过 3 个心跳周期未收到心跳，建议重启注册鉴权流程。

### 7.3.2 处理时延

在硬件设施及网络正常的前提下，算法处理事件的时延应小于1秒。